The non-adaptive query complexity of testing k-parities

Harry Buhrman* David García-Soriano Arie Matsliah Ronald de Wolf†

September 19, 2012

Abstract

We prove tight bounds of $\Theta(k \log k)$ queries for non-adaptively testing whether a function $f : \{0,1\}^n \to \{0,1\}$ is a k-parity or far from any k-parity. Both upper and lower bounds are new. The lower bound combines a recent method of Blais, Brody and Matulef [BBM11] to get testing lower bounds from communication complexity, with a new $\Theta(k \log k)$ bound for the one-way communication complexity of k-disjointness.

1 Introduction

A *parity* is a function $f : \{0,1\}^n \to \{0,1\}$ that can be written as $f(y) = \langle x, y\rangle$, the inner product (mod 2) of y with some fixed string x. We also sometimes denote this function by $f = x^*$. We call f a *k-parity* if x has Hamming weight k. We consider the following testing problem:

> Let $1 \le k \le n$ be integers. Given oracle access to a Boolean function $f : \{0,1\}^n \to \{0,1\}$, how many queries to f do we need to test (i.e., determine with probability $\ge 2/3$) whether f is a k-parity or far from any k-parity?

Here a function f is *far* from a set of functions G, if for all $g \in G$, the functions f and g differ on at least a constant fraction of their domain $\{0,1\}^n$ (for concreteness one can take this constant to be $1/10$). Let PAR^n_k denote the set of all k-parities on n-bit inputs, $\mathsf{PAR}^n_{\le k} = \cup_{\ell \le k} \mathsf{PAR}^n_\ell$, and $\mathsf{PAR}^n = \mathsf{PAR}^n_{\le n}$.

Another way of looking at the problem is as determining, by making as few queries as possible to the *Hadamard encoding* of a word x, whether $|x| = k$ or not. So the task is essentially how to decide if $|x| = k$ efficiently if we can query the XOR of arbitrary subsets of the bits of x.[1]

It is easy to see that deciding if the size of a parity is k is the same problem as deciding if it is $n - k$. For even n, the case $k = n/2$ is particularly interesting because it enables us to verify the equality between the sizes of two unknown parities $f, g \in \mathsf{PAR}^n$. Indeed, define a parity on $2n$ variables by $h(x_1 x_2) = f(1^n \oplus x_1) \oplus g(x_2)$, where $x_1, x_2 \in \{0,1\}^n$; then $h \in \mathsf{PAR}^{2n}_n$ if and only if f and g are parities of the same size.

A related problem is deciding if a parity has size *at most* k (naturally, this is equivalent to deciding if the size is at least $n - k$, or at most $n - k - 1$). Upper bounds for this task imply

*Partially supported by the European Commission under the project QCS (Grant No. 255961).

†Partially supported by a Vidi grant from the Netherlands Organization for Scientific Research (NWO), and by the European Commission under the project QCS (Grant No. 255961).

[1] Decision trees where the queries are allowed to be XORs of subsets of the inputs have appeared in the literature [ZS10].

upper bounds for testing k-parities (one can perform one test to verify the condition $|x| \leq k$ and another one for $|x| \leq k-1$). Lower bounds here do not immediately imply lower bounds for testing isomorphism, but they (and so lower bounds for testing k-parities) do imply lower bounds for testing k-juntas (because one way of checking if $f \in \mathsf{PAR}^n_{\leq k}$ is testing that f is linear and also a k-junta).

The first step towards analyzing the hardness of these problems was taken by Goldreich [Gol10, Theorem 4], who proved that testing if a linear function $f \in \mathsf{PAR}^n$ (n even) is in $\mathsf{PAR}^n_{\leq n/2}$ requires $\Omega(\sqrt{n})$ queries. Goldreich conjectured that the true bound should be $\Theta(n)$. Later Blais et al. [BBM11] showed that testing if a function f is a k-parity requires $\Omega(k)$ queries.

In this paper we focus on *non-adaptive* testing, where all queries to f are chosen in advance. Our main results are tight upper and lower bounds of $\Theta(k \log k)$ non-adaptive queries for testing whether f is in or far from the set PAR^n_k. Section 2 describes our upper bound and Section 3 describes our lower bound, which is based on a new (and tight) $\Theta(k \log k)$ bound for the one-way communication complexity of the k-disjointness problem. After obtaining our results, we learned that this same communication complexity result has also independently been obtained by Dasgupta, Kumar and Sivakumar [DKS12].

2 Upper bounds

Here we prove that $O(k \log k)$ queries suffice to non-adaptively test if $f : \{0,1\}^n \to \{0,1\}$ is a k-parity. We assume $k = \omega(1)$, as the result is easy to establish if $k = O(1)$. First we show a tester for the special case $n = 100k^2$, and then we show how the general case reduces to this special case.

The basic ingredient we need is the *influence test* (see also [FKR$^+$04]).

Claim 1. Influence test *Let $f : \{0,1\}^n \to \{0,1\}$ be a parity function with $J \subseteq [n]$ being the set of its influential variables. There is a probabilistic procedure $I_f : \{0,1\}^n \to \{0,1\}$ that when executed on input $x \in \{0,1\}^n$ (corresponding to a set $x \subseteq [n]$) satisfies the following:*

- *I_f makes at most 7 queries to f;*
- *if $x \cap J = \emptyset$ then I_f returns 0;*
- *if $x \cap J \neq \emptyset$ then I_f returns 1 with probability at least 99/100.*

In other words, $I_f(x)$ is a probabilistic predicate (with one-sided error) checking if x and J intersect.

The influence test can be made more robust, to handle functions f that are only close to being parities, by increasing the query-complexity (per test) and switching to two-sided error:

Claim 2. Noisy influence test *Let $f : \{0,1\}^n \to \{0,1\}$ be 1/10-close to a parity function $g : \{0,1\}^n \to \{0,1\}$ with influential variables $J \subseteq [n]$. There is a probabilistic procedure $I^N_f : \{0,1\}^n \to \{0,1\}$ that when executed on input $x \in \{0,1\}^n$ satisfies the following:*

- *I^N_f makes at most 210 queries to f;*
- *if $x \cap J = \emptyset$ then I^N_f returns 0 with probability at least 49/50;*
- *if $x \cap J \neq \emptyset$ then I^N_f returns 1 with probability at least 49/50.*

In other words, $I_f^N(x)$ is a probabilistic predicate checking if x and J (the influential variables of the parity function g closest to f) intersect.

Proof. We use the self-correction property of the Hadamard code:

$$\Pr_{y\in\{0,1\}^n}[g(x) = f(y) \oplus f(y \oplus x)] \geq 1 - 2 \cdot \mathrm{dist}(f, g) \geq 4/5.$$

This allows us to correctly decode the value of g on any given input with probability $1 - 1/700$ using 30 queries. Hence, by the union bound, any 7 values (for a single application of the usual influence test) can be decoded correctly with probability $1/100$. Now use the tester from Claim 1 and observe that the overall error probability is at most $1/100 + 1/100 = 1/50$. □

So, prior to testing if f is a k-parity, we test it for being a parity function with proximity parameter $1/10$ and confidence parameter $99/100$ (this can be done with a constant number of queries [BLR90]). If this test fails then we reject; otherwise, we assume f is $1/10$-close to being a parity function and condition all further probabilities accordingly.

2.1 Testing in the case where $n = 100k^2$

In the following test we set $q = \frac{1000}{\rho} k \log k$, with $\rho \in (0, 1]$ being a constant defined later.

- Draw $r_1, \ldots, r_q \in \{0,1\}^n$ at random, by setting r_{ij} to 1 with probability ρ/k for each $i \in [q]$ and $j \in [n]$, independently of the others. For each $j \in [n]$, denote by $S^j \subseteq [q]$ the set of indices $i \in [q]$ with $r_{ij} = 1$.
- Compute $a_i \leftarrow I_f^N(r_i)$ for all $i \in [q]$ with the noisy influence test of Claim 2. For each $j \in [n]$ denote by S_1^j the subset of S^j containing indices i with $a_i = 1$.
- Output the subset $\hat{J} \subseteq [n]$ containing the indices j for which $|S_1^j| > \frac{3}{4}|S^j|$, and *accept* if and only if $|\hat{J}| = k$.

The next claim says that with high probability, all influential variables of f^* (the parity function closest to f) are inserted in $\hat{J}$.

Claim 3. *With probability $1 - o(1)$ the following conditions are simultaneously satisfied:*

- $|S^j| > 100 \log k$ *for every* $j \in [n]$*;*
- $|S_1^j| > \frac{3}{4}|S^j|$ *for every* $j \in J$*.*

Proof. Apply standard concentration bounds to prove the first item (note that the expectation of $|S^j|$ is $1000 \log k$). Then, conditioned on it, use Claim 2 and another application of a concentration bound to get the second item. □

The next claim says that when $|J| \leq k$, with high probability none of the non-influential variables are inserted in $\hat{J}$. Before we proceed, let us call an index $i \in [q]$ *intersecting* w.r.t. J if $r_i \cap J \neq \emptyset$. Recall that the probability of any one element of J belonging to r_i is ρ/k; therefore the probability that i is *not* intersecting is $(1 - \rho/k)^{|J|} \geq (1 - \rho/k)^k$. We set the constant ρ so that for a fixed J of size at most k, the probability that i is intersecting is at most $1/10$, for each i.

Claim 4. *If $|J| \leq k$, then with probability $1 - o(1)$ the following conditions are simultaneously satisfied:*

- *$|S^j| > 100 \log k$ for every $j \in [n]$;*
- *for every $j \notin J$, the fraction of non-intersecting indices in $|S^j|$ is $> 1/2$.*

Here, too, the proof follows by straightforward application of standard concentration bounds.

To conclude the correctness of the tester, observe that by Claim 3, with high probability $J \subseteq \hat{J}$, so if J contains more than k indices, then so will $\hat{J}$. On the other hand, if $|J| \leq k$ then by Claim 4, with high probability all sets S^j with $j \notin J$ contain a majority of non-intersecting indices. For each non-intersecting $i \in S^j$, it holds that a_i is 0 with probability $\geq 49/50$. But in order for j to belong to $\hat{J}$, at least three quarters of the indices $i \in S^j$ must have $a_i = 1$, which implies that at least half of the non-intersecting indices i of S^j must have $a_i = 1$. As there are at least $|S^j|/2 > 50 \log k$ non-intersecting indices in S^j, standard concentration estimates show that this happens with probability at most k^{-c} for some $c > 2$. Since we are in the case $n = 100k^2$, this probability is $o(1/n)$ for each $j \in [n]$, and we can apply the union bound to conclude that the success probability of the tester is $1 - o(1)$.

Note that the test does more than testing: it actually *identifies* the set J of influential variables as long as it is of size $\leq k$.

2.2 Reducing the general case to $n = 100k^2$

Lemma 5. *Let $k > 100$ and $n > 100k^2$. Given a subset $J \subseteq [n]$ and a $100k^2$-way partition $\Pi = S_1, \ldots, S_{100k^2}$ of $[n]$, we denote by $N(\Pi, J)$ the number of classes S_i containing an odd number of elements from J. The following holds for randomly constructed partitions Π:*

- *for each $J \subseteq [n]$ of size $|J| \leq k$, $\Pr_\Pi[N(\Pi, J) = |J|] > 9/10$,*
- *for each $J \subseteq [n]$ of size $|J| > k$, $\Pr_\Pi[N(\Pi, J) > k] > 9/10$.*

Proof. Assume $n > 100k^2$ (otherwise the trivial partition with singleton and empty classes satisfies both conditions). If $|J| \leq k$, then by a birthday-paradox type argument, with probability $\geq 9/10$ no pair of indices from J belong to the same partition class, and hence $N(\Pi, J) = |J|$.

Now let $|J| > k$. Consider the stage in the construction of the random partition Π where all but the last $k+1$ elements from J were mapped to one of Π's classes. If at this stage $N(\Pi, J) > 2k+2$ then we are done (since adding $k+1$ indices from J can only change $N(\Pi, J)$ by $k+1$). Otherwise, we use a birthday-paradox type argument again to show that with probability 9/10 no pair from a set of $\leq 3k+2$ elements collides when randomly mapped to $100k^2$ classes. □

Once such a partition is obtained[2] we can simulate access to a function $f' : \{0,1\}^{100k^2} \to \{0,1\}$ by querying f on inputs that are constant within each partition class, and reduce the original problem of testing f to the problem of testing whether f' is a k-parity.

Putting everything together, we have proved our upper bound:

Theorem 6. *There exists a non-adaptive tester that uses $O(k \log k)$ queries to a given function $f : \{0,1\}^n \to \{0,1\}$, and decides with probability at least $2/3$ whether f is in or far from PAR_k^n.*

[2] We condition all calculations in Section 2.1 on this event, which occurs with probability 9/10.

3 Lower bounds

3.1 The one-way communication complexity of k-disjointness

In two-party communication complexity [Yao79, KN97], two parties (Alice and Bob) have inputs x and y, respectively, and want to compute some function of x and y. Unlimited access to their respective inputs and arbitrary computations are allowed, and the measure for the protocol's efficiency is the number of bits of communication they need to transmit to each other. We consider the model where Alice and Bob are share a common source of randomness ("public coin') and are allowed to err with probability at most 1/3.

In the *k-disjointness* problem, Alice and Bob receive two k-sets $x, y \in \binom{[n]}{k}$ and would like to determine if $x \cap y = \emptyset$ or not. Furthermore, they are guaranteed that either $x \cap y = \emptyset$ or $|x \cap y| = 1$. This problem is known to have communication complexity $\Theta(k)$. The upper bound is due to Håstad and Wigderson [HW07], the lower bound due to Kalyanasundaram and Schnitger, and subsequent simplifications and strengthenings were found by Razborov [Raz92] and Bar-Yossef et al. [BJKS04]. The Håstad-Wigderson protocol is interactive (i.e., it uses many rounds of communication), and we show here this is actually necessary: if we just allow one-way communication from Alice to Bob, then the lower bound goes up from $\Omega(k)$ to $\Omega(k \log k)$ bits.

Theorem 7. *The one-way communication complexity of the k-disjointness problem is* $\Theta(k \log k)$ *for* $k \leq \sqrt{n/2}$, *and* $\Theta(\log \binom{n}{k})$ *for* $k > \sqrt{n/2}$.

Proof. For the upper bound, first note that Alice can just send Bob the index of her input x in the set of all weight-k strings of length n, at the expense of $\log \binom{n}{k}$ bits. If $k \leq \sqrt{n/2}$ then we can do something better, as follows. Alice and Bob use the shared randomness to choose a random partition of their inputs into $b = O(k^2)$ buckets, each of size n/b. By similar birthday paradox arguments as before, with probability close to 1 no two 1-positions in x will end up in the same bucket, and no two 1-positions in y will end up in the same bucket. We condition the remainder of the upper-bound argument on this successful bucketing. Note that x and y intersect iff there is an $i \in [b]$ such that Alice and Bob's strings in the ith bucket are equal and non-zero. For each of her k non-empty buckets, Alice sends Bob the index of that bucket, and uses the well-known public-randomness equality protocol on that bucket: they choose $2 \log k$ uniformly random strings $r_1, \ldots, r_{2 \log k} \in \{0,1\}^{n/b}$ and Alice sends over the inner products (mod 2) of her bucket with each of those strings. Bob compares the bits he received with the inner products of $r_1, \ldots, r_{2 \log k}$ with his corresponding bucket. If their two buckets are the same then all inner products will be the same, and if their two buckets differ in at least one bit-position then they will see a difference in those inner products, except with probability $1/2^{2 \log k} = 1/k^2$. Bob checks whether one of Alice's non-empty buckets equals his corresponding bucket. If so he concludes that x and y intersect, and otherwise he concludes that they are disjoint. Taking the union bound over the probability that the bucketing fails and the probability that one of the k equality tests fails, shows that the error probability is close to 0. The communication cost of this one-way protocol is $O(\log k)$ bits for each of Alice's non-empty buckets, so $O(k \log k)$ bits in total.

For the lower bound, first consider the case $k \leq \sqrt{n/2}$. Let x be Alice's input, viewed as an n-bit string of Hamming weight k. For Alice we restrict our attention to inputs of a particular structure. Namely, partition $[n]$ into k consecutive sets of size $n/k \geq 2k$. The inputs we allow contain precisely one bit set to 1 inside each block of the partition, and moreover the offset of the unique index set to one within the ith block is an integer in $\{0, \ldots, 2k-1\}$. In this case, x describes a message M

of k integers $m_1, \ldots, m_k$, each in the interval $\{0, \ldots, 2k-1\}$. M can also be viewed as an m-bit long message, where $m = k \log(2k)$. We can write Alice's input as $x = u(m_1) \ldots u(m_k)$, where $u(m_i) \in \{0,1\}^{n/k}$ is the unary expression of the number m_i using n/k bits (where the rightmost $n/k - k$ bits of each $u(m_i)$ are always zero). For instance, the picture below illustrates the case where $n = 40$, $k = 4$, and $M = (1, 7, 0, 5)$:

$$x = \underbrace{\overbrace{0100000000}^{n/k}}_{u(m_1)} \ \underbrace{\overbrace{0000000100}^{n/k}}_{u(m_2)} \ \underbrace{\overbrace{1000000000}^{n/k}}_{u(m_3)} \ \underbrace{\overbrace{0000010000}^{n/k}}_{u(m_4)}$$

Let ρ_x be the q-bit message that Alice sends on this input; this is a random variable, depending on the public coin. Below we show that the message is a *random-access code* for M, i.e., it allows a user to recover each bit of M with probability at least $1 - \delta$ (though not necessarily *all* bits of M simultaneously). Then our lower bound will follow from Nayak's random-access code lower bound [Nay99]. This says that

$$q \geq (1 - H(\delta))m,$$

where δ is the error probability of the protocol and $H(\delta) = -\delta \log(\delta) - (1-\delta) \log(1-\delta)$ is its binary entropy.

Suppose Bob is given ρ_x and wants to recover some bit of M. Say this bit is the ℓth bit of the binary expansion of m_i. Then Bob completes the protocol using the following y: y is 0 everywhere except on the k bits in the ith block of size n/k whose offsets j (measured from the start of the block) satisfy the following: $0 \leq j < 2k$ and the ℓth bit of the binary expansion of j is 1. The Hamming weight of y is k by definition.

Recall that Alice has a 1 in block i only at position m_i. Hence x and y will intersect iff the ℓth bit of the binary expansion of m_i is 1, and moreover, the size of the intersection is either 0 or 1. Running the k-disjointness protocol with confidence $1 - \delta$ will now give Bob the sought-for bit of M with probability at least $1 - \delta$, which shows that ρ_x is a random-access code for M.

If $k > \sqrt{n/2}$ then we can do basically the same lower-bound proof, except that the integers m_i are now in the interval $\{0, \ldots, n/k - 1\}$, $m = k \log(n/k)$, and Bob puts only $n/2k < k$ ones in the ith block of y (he can put his remaining $k - n/2k$ indices somewhere at the end of the block, at an agreed place where Alice won't put 1s). This gives a lower bound of $\Omega(k \log(n/k)) = \Omega(\log \binom{n}{k})$. □

We note that the lower bound holds even for *quantum* one-way communication complexity, and even if we allow Alice and Bob to share entanglement. For the latter case Nayak's random access-code lower bound [Nay99] needs to be replaced with Klauck's [Kla00] version, which is weaker by a factor of two.

3.2 Non-adaptive lower bound for testing k-parities

In a recent paper, Blais, Brody and Matulef [BBM11] made a clever connection between property testing and some well-studied problems in communication complexity. As one of the applications of this connection, they used the $\Omega(k)$ lower bound for k-disjointness to prove an $\Omega(k)$ lower bound on testing whether a function is in or far from the class of k-parities. We use their argument to get a better lower bound for *non-adaptive* testers:

Corollary 8. *Let $1 \leq k \leq n$. If $k \leq \sqrt{n/2}$ then non-adaptive testers need at least $\Omega(k \log k)$ queries to test with success probability at least $2/3$ whether a given function $f : \{0,1\}^n \to \{0,1\}$ is in or far from PAR_k^n; and if $k > \sqrt{n/2}$ then they need at least $\Omega(\log \binom{n}{k})$ queries.*

Proof. Let k be even (a similar argument works for odd k). Below we show how Alice and Bob can use a non-adaptive q-query tester for k-parities to get a one-way public-coin communication complexity for $k/2$-disjointness with q bits of communication. The communication lower bound of Theorem 7 then implies the result.

Alice forms the function $f = x^*$ and Bob forms the function $g = y^*$. Consider the function $h = (x \oplus y)^*$. Since $|x \oplus y| = |x| + |y| - 2|x \cap y|$, the function h is a k-parity if $x \cap y = \emptyset$, and a $(k-2)$-parity if $|x \cap y| = 1$. A q-query randomized tester is a probability distribution over q-query deterministic testers. Alice and Bob use the public coin to jointly sample one of those deterministic testers. Since the tester is non-adaptive, this fixes the q queries that will be made. For every such query $z \in \{0,1\}^n$, Alice sends Bob the bit $f(z)$. This enables Bob to compute $h(z) = f(z) \oplus g(z)$ for all q queries and then to finish the computation of the tester. Since a $(k-2)$-parity has distance $1/2$ from every k-parity, the tester will tell Bob whether h is a k-parity or a $(k-2)$-parity; equivalently, whether x and y intersect or not. □

As mentioned in the introduction, a lower bound for testing membership in PAR_k^n implies a lower bound for $\mathsf{PAR}_{\leq k}^n$ and juntas.

4 Conclusion and future work

We end with a few comments and directions for future research:

- While our disjointness lower bound (Theorem 7) also applies to one-way *quantum* protocols, our lower bound for testing (Corollary 8) does not. The reason is that the overhead when turning a quantum tester into a communication protocol will be $O(n)$ qubits per query in the quantum case, in contrast to the $O(1)$ bits per query in the classical case. In fact, if f is a k-parity then the Bernstein-Vazirani algorithm [BV97] finds x itself using only one quantum query, so testing for k-parities is trivial for quantum algorithms.

- For *adaptive* testers for k-parities there is still a gap between the best lower bound of $\Omega(k)$ queries and the best upper bound of $O(k \log k)$ queries. It would be interesting to close this gap.

Acknowledgements

We thank Pavel Pudlak for asking a question (communicated via HB) related to lower bounds for monotone circuits and span programs, which prompted RdW to prove the lower bound part of Theorem 7 in October 2006; this proof was unpublished until now. We thank Joshua Brody for referring us to [DKS12], and Anirban Dasgupta for sending us a copy of that paper.

References

[BBM11] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *Proc. 26th CCC*, 2011.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci*, 68:702–732, June 2004.

[BLR90] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proc. 22nd STOC*, pages 73–83, 1990.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[DKS12] Anirban Dasgupta, Ravi Kumar, and D. Sivakumar. Sparse and lopsided set disjointness via information theory. In *Proc. 16th RANDOM*, 2012. To appear.

[FKR$^+$04] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci*, 68(4):753–787, 2004.

[Gol10] Oded Goldreich. On testing computability by small width OBDDs. In *Proc. 14th RANDOM*, pages 574–587, 2010.

[HW07] Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007.

[Kla00] Hartmut Klauck. On quantum and probabilistic communication: Las Vegas and one-way protocols. In *Proc. 32nd STOC*, pages 644–651, 2000.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[Nay99] Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proc. 40th FOCS*, pages 369–376, 1999.

[Raz92] Alexander Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106:385–390, December 1992.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *Proc. 11st STOC*, pages 209–213, 1979.

[ZS10] Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theoretical Computer Science*, 411:2612–2618, June 2010.